NASA Procedural Requirements

NPR 7150.2A

Effective Date:
November 19, 2009
Expiration Date:
November 19, 2014

COMPLIANCE IS MANDATORY

Printable Format (PDF)

Request Notification of Change   (NASA Only)

Subject: NASA Software Engineering Requirements

Responsible Office: Office of the Chief Engineer

Chapter 4: Supporting Software Life-Cycle Requirements

Support processes are not limited to a single software life-cycle phase such as requirements, design, implementation, or test. Support processes typically occur throughout the software life cycle. For example, typical configuration management baselines (e.g., requirements, code, products) happen across the life cycle. Support processes are software management and engineering processes that typically support the entire software life cycle (e.g., configuration management).

4.1 Software Configuration Management

Software configuration management is the process of applying configuration management throughout the software life cycle to ensure the completeness and correctness of software configuration items. Software configuration management applies technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of software configuration items, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. Software configuration management establishes and maintains the integrity of the products of a software project throughout the software life cycle. Use of standard Center or organizational software configuration management processes and procedures is encouraged where applicable.

4.1.1 The project shall develop a Software Configuration Management Plan that describes the functions, responsibilities, and authority for the implementation of software configuration management for the project. [SWE-079]

Note: The Software Configuration Management Plan may be a part of the project configuration management plan. The content is defined by the requirement in Chapter 5.

4.1.2 The project shall track and evaluate changes to software products. [SWE-080]

Note: The project can use a software change request system or a software problem tracking system. The minimum content for software change requests or a software problem report is defined in Chapter 5.

4.1.3 The project shall identify the software configuration items (e.g., software documents, code, data, tools, models, scripts) and their versions to be controlled for the project. [SWE-081]

Note: The project is responsible for assuring that software safety elements are properly identified and controlled.

4.1.4 The project shall establish and implement procedures designating the levels of control each identified configuration item must pass through; the persons or groups with authority to authorize changes and to make changes at each level; and the steps to be followed to request authorization for changes, process change requests, track changes, distribute changes, and maintain past versions. [SWE-082]

4.1.5 The project shall prepare and maintain records of the configuration status of configuration items. [SWE-083]

Note: Configuration status accounting generates and/or maintains records of the status and contents of the software throughout the life cycle. This function keeps track of the changes and the contents of versions and releases.

4.1.6 The project shall ensure that software configuration audits are performed to determine the correct version of the configuration items and verify that they conform to the documents that define them. [SWE-084]

4.1.7 The project shall establish and implement procedures for the storage, handling, delivery, release, and maintenance of deliverable software products. [SWE-085]

4.2 Risk Management

Identification and management of risks provide a basis for systematically examining changing situations over time to uncover and correct circumstances that impact the ability of the project to meet its objectives.

4.2.1 The project shall identify, analyze, plan, track, control, communicate, and document software risks in accordance with NPR 8000.4, Agency Risk Management Procedural Requirements. [SWE-086]

Note: A project needs to include an assessment of the risk that any untested code poses to the system or subsystem.

4.3 Software Peer Reviews/Inspections

Software peer reviews and inspections are the in-process technical examination of work products by peers to find and eliminate defects early in the life cycle. Software peer reviews/inspections are performed following defined procedures covering the

preparation for the review, conducting the review itself, documenting results, reporting the results, and certifying the completion criteria. When planning the composition of a software peer review/inspection team consider including software testing, system testing, software assurance, software safety, and software Independent Verification and Validation (IV&V) personnel.

4.3.1 The project shall perform and report on software peer reviews/inspections for: [SWE-087]

a. Software requirements.

b. Software Test Plan.

c. Any design items that the project identified for software peer review/inspections according to the software development plans.

d. Software code as defined in the software and or project plans.

Note: Software peer reviews/inspections are a recommended best practice for all safety and mission-success related design and code software components. Guidelines for software peer reviews/inspections are contained in NASA-STD-2202-93, NASA Software Formal Inspection Standard.

4.3.2 The project shall perform and report on software peer reviews/inspections for: [SWE-137]

a. Software Development or Management Plan.

b. Software Configuration Management Plan.

c. Software Maintenance Plan.

d. Software Assurance Plan.

e. Software Safety Plan.

4.3.3 The project shall, for each planned software peer review/inspections: [SWE-088]

a. Use a checklist to evaluate the work products.

b. Use established readiness and completion criteria.

c. Track actions identified in the reviews until they are resolved.

d. Identify required participants.

4.3.4 The project shall, for each planned software peer review/inspection, record basic measurements. [SWE-089]

Note: The requirement for the content of a Software Peer Review/Inspection Report is defined in Chapter 5.

4.4 Software Measurement

Software measurement programs at multiple levels are established to meet measurement objectives. The requirements below are designed to establish measurement programs at the project and the Mission Directorate levels to assist in managing projects, assuring quality, and improving software engineering practices.

Project-level and Mission Directorate/Mission Support Office-level (product line) measurement programs are designed to meet the following high-level goals:

a. To improve future planning and cost estimation.

b. To provide realistic data for progress tracking.

c. To provide indicators of software quality.

d. To provide baseline information for future process improvement activities.

Additional measures can be defined by either the projects or the Mission Directorate/Mission Support Office, based on any additional high-level goals they may have.

4.4.1 The project shall establish and document specific measurement objectives for their project. [SWE-090]

4.4.2 The project shall select and record the selection of specific measures in the following areas: [SWE-091]

a. Software progress tracking.

b. Software functionality.

c. Software quality.

d. Software requirements volatility.

e. Software characteristics.

Note: The requirement for a Software Metrics Report is defined in Chapter 5.

4.4.3 The project shall specify and record data collection and storage procedures for their selected software measures and collect and store measures accordingly. [SWE-092]

4.4.4 The project shall analyze software measurement data collected using documented project-specified and Center/organizational analysis procedures. [SWE-093]

4.4.5 The project shall report measurement analysis results periodically and allow access to measurement information by Center-defined organizational measurement programs. [SWE-094]

4.4.6 Each NASA Mission Directorate shall establish its own software measurement system to include the minimum reporting requirements in SWE-091. [SWE-095]

4.4.7 Each NASA Mission Directorate shall identify and document the specific measurement objectives, the chosen specific measures, the collection procedures, and storage and analysis procedures. [SWE-096]

[Requirement number SWE-097 is reserved]

4.5 Best Practices

Best practices that are collected by the software community are a resource for improving software products. Ensuring an awareness of these practices can often provide potential solutions to problems. Successful best practices also provide alternate approaches for

an individual project to consider, given its scope, domain, and goals. The intent of organizational best practices is not to mandate the use of any specific practice, but to provide information and examples to each project so that it can evaluate and choose those practices that it deems most beneficial. The identified best practices can then be used to improve the life-cycle processes for future software product improvement purposes.

4.5.1 The NASA Headquarters' Office of the Chief Engineer shall maintain an Agency-wide process asset library of applicable best practices. [SWE-098]

Note: The Agency-level process assets can be viewed from the NASA Software Process Asset Library (PAL) Web site, http://swpal.nasa.gov/, from the NASA Software Engineering Web site at http://software.nasa.gov and from the NASA Engineering Network Web site at http://nen.nasa.gov/portal/site/llis/OCE/. The repository may contain information in many forms including, but not limited to, processes, Web sites, design principles, books, periodicals, presentations, tools, examples of documents, and conference descriptions.

4.5.2 Each Center shall review the contents of the process asset library to identify those practices that may have direct applicability and value to its software activities. [SWE-099]

4.6 Training

Having properly trained personnel is one of the key items that can lead to success of software engineering projects. The goal is to maintain and advance organizational capability for training of personnel that perform software engineering practices to effectively meet scientific and technological objectives. The Software Training Plan includes training in the following software activities: software management, software acquisition, software monitoring, software development, software safety and mission assurance, and software process improvement.

4.6.1 The NASA Headquarters' Office of the Chief Engineer and Center training organizations shall provide and fund training to advance software engineering practices and software acquisition. [SWE-100]

4.6.2 Each Center shall maintain and implement Software Training Plan(s) to advance its in-house software engineering capability and as a reference for its contractors. [SWE-101]

Note: The Software Training Plan content is defined by the Software Training Plan requirement in Chapter 5.

| TOC | Preface | Chapter1 | Chapter2 | Chapter3 | Chapter4 |
Chapter5 | Chapter6 | AppendixA | AppendixB | AppendixC |
AppendixD | AppendixE | ALL |

| NODIS Library | Program Formulation(7000s) | Search |

NPR 7150.2A -- Chapter4

Verify Current version before use at:
http://nodis3.gsfc.nasa.gov/

Page 5 of 6

DISTRIBUTION:
NODIS